

# A P2P Technology Based On Relational Database In Cloud Computing

<sup>#1</sup>Akshata Desai, <sup>#2</sup>Zeenat Kalambkar, <sup>#3</sup>Uma Madake, <sup>#4</sup>Megha Jadhav

<sup>1</sup>akshatadesai7777@gmail.com

<sup>2</sup>kzeenu20@gmail.com

<sup>3</sup>madkeuma02@gmail.com

<sup>4</sup>meghadjadhav004@gmail.com

<sup>#1234</sup>Department of Computer Engineering

JSPM's

Bhivarabai Sawant College of Engineering & Research,  
Pune, India.



## ABSTRACT

A corporate network is a group of computers, connected together in a building or in a particular area, which are all owned by the same company or institutions. It provides the flexible, economical & Scalable platform. Data management system including scalability, flexibility, security, Data Sharing, better performance, no of peers easily added & remove it's challenging for corporate network. Centralized data processing where data processing supported by one clusters of computers means all data stored on centralized platform. In centralized data processing it's necessary to handle the computers overload problem; this problem is overcome in using Best Peer++ concept. Best peer ++ achieves linear scalability throughput with respect to the number of peer nodes, this problem is overcome using bootstrap and adaptive query processing algorithm.

**Keywords :** Corporate network, Cloud computing, Peer to peer system, HadoopDB, Query Processing .

## ARTICLE INFO

### Article History

Received: 11<sup>th</sup> October 2015

Received in revised form :

11<sup>th</sup> October 2015

Accepted : 14<sup>th</sup> October 2015

**Published online :**

15<sup>th</sup> October 2015

## I. INTRODUCTION

Data engineers are the designers, builders and managers of the information or "big data" infrastructure. They develop the architecture that helps analyze and process data in the way the organization needs it. Such a warehousing solution has some deficiencies in real deployment. First, the corporate network needs to scale up to support thousands of participants, while the installation of a large-scale centralized data warehouse system entails nontrivial costs including huge hardware/software investments (total cost of ownership) and high maintenance cost (total cost of operations). In the real world, most companies are not keen to invest heavily on additional information systems until they can clearly see the potential return on investment (ROI). Second, companies want to fully customize the access control policy to determine which business partners can see which part of their shared data.

The main contribution of this paper is the design of BestPeer++ system that provides economical, flexible and scalable solutions for corporate network applications. We demonstrate the efficiency of BestPeer++ by benchmarking BestPeer++ against HadoopDB, a recently proposed large-scale data processing system, over a set of queries designed for data sharing applications. The results show that for

simple, low-overhead queries, the performance of BestPeer++ is significantly better than HadoopDB. The unique challenges posed by sharing and processing data in an inter-businesses environment and proposed BestPeer++, a system which delivers elastic data sharing services, by integrating cloud computing, database, and peer-to-peer technologies.

### A) Problem Statement:

From a technical perspective for the success of a corporate network is choosing the right data sharing platform which is a system which enables the shared data (stored and maintained by different companies) network-wide visible and supports efficient analytical queries over those data. Most of the times data sharing is achieved by building a centralized data warehouse. But the unique challenges to such a data management system such as scalability and performance.

## II. LITERATURE SURVEY

In To enhance the usability of conventional peer to peer systems database communities have proposed a series of PDBMS (Peer-to-Peer Database Manage System) by

integrating the state-of-art database techniques into the P2P systems. There are many techniques proposed in order to efficiently process large scale data which has explained below:

A) S. Wu and J. Li have proposed “Just-in-Time Query Retrieval over Partially Indexed Data on Structured P2P Overlays,”

It is a Peer-based system that Indexes Selected Content for Efficient Search facility. It is not like traditional approaches that index all data and PISCES identifies a subset of tuples to index based on some criteria. Another important addition to this is a coarse-grained range index is built to facilitate the processing of queries that cannot be fully answered by the tuple-level index. The main limitation is the possibly high maintenance cost to maintain the structure

B) K.-L. Tan and A. Zhou presented “PeerDB: A P2P-Based System for Distributed Data Sharing,”

PeerDB is a peer to peer based database management System which employs information retrieval technique to match columns of different tables. The main problem of unstructured PDBMS is that there is no guarantee for the data retrieval performance and result quality.

C) S. Jiang and B.C. Ooi have proposed “Distributed Online Aggregation,”

In this paper they extend the on-line aggregation technique to a distributed context where sites are maintained in a DHT (Distributed Hash Table) network. Distributed Online Aggregation (DoA) scheme iteratively and progressively produces approximate aggregate answers as follows 1) in each iteration small set of random samples are retrieved from the data sites and distributed to the processing sites 2) at each processing site local aggregate is computed based on the allocated samples and 3) at a coordinator site these local aggregates are combined into a global aggregate.

D) A. Lakshman and A. Pilchin “Dynamo: Amazon’s Highly Available Key-Value Store”

This paper presents the design and implementation of Dynamo which is a highly available key-value storage system that some of Amazon’s core services use to provide an always-on experience. The important thing here is that it makes extensive use of object versioning and application-assisted conflict resolution in a manner that provides a novel interface for developers to use.

### III. PROPOSED SYSTEM

The original BestPeer system attempts to exploit peer-to-peer (P2P) technologies for distributed applications. BestPeer was designed to work as a scalable, sharable, and secure P2P-based Data Management system with full functionalities for building corporate networks in which a group of organizations controlled by different administrative domains collaborate with each other in order to reduce operation cost and improve productivity. We propose a balanced tree (BATON tree) structure overlay on a peer-to-peer network capable of supporting both exact queries and range queries efficiently. In spite of the tree structure causing distinctions to be made between nodes at different levels in the tree, we show that the load at each node is approximately equal. In spite of the tree structure providing precisely one path between any pair of nodes, we show that

sideways routing tables maintained at each node provide sufficient fault tolerance to permit efficient repair. We are implementing our project by using Java Technology and MySQL database.

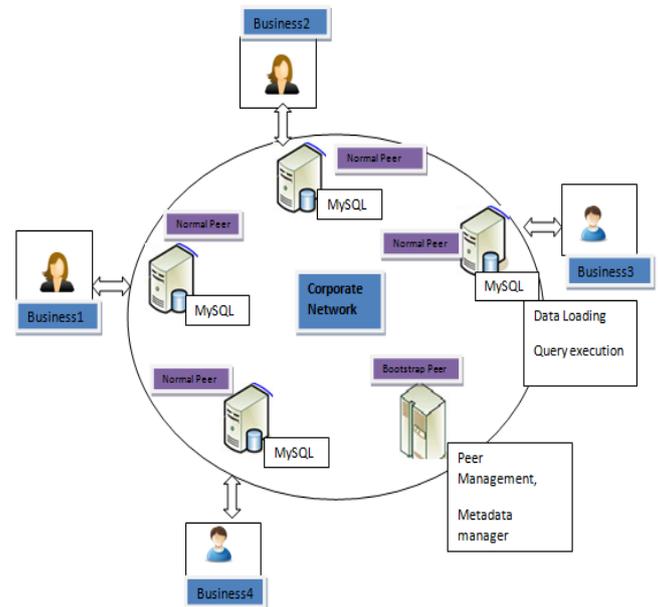


Fig 1. System Architecture

Various components of our system are:

#### A.) Corporate network

Corporate network is used for sharing information among the participating companies and facilitating collaboration in a certain industry sector where companies share a common interest among them.

We are creating here a corporate network where each business registers in the corporate network. The network service provider saves this registration information and allows each business or company to share their data with other companies present in the network. The businesses can then upload their data to their local databases. And allow access to this data to corporate network companies.

#### B.) Bootstrap Peer

The bootstrap peer is run by the BestPeer++ service provider. The important functionality of this peer is to manage the BestPeer++ network. Every normal peer or node wants to join an existing corporate network must first connect to the bootstrap peer. The bootstrap peer authenticates this information. If the join request is permitted by the service provider, the bootstrap peer will put the newly joined peer into the peer list of the corporate network. In addition to managing peer join and peer departure another functionality of bootstrap peer is to monitoring the health of normal peers and scheduling fail-over and auto-scaling events. In this function the bootstrap periodically collects performance metrics of each normal peer.

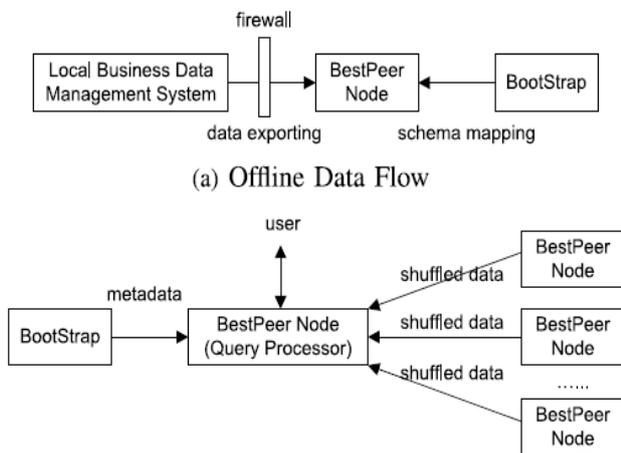
**Bootstrap Daemon Algorithm**

```

While true do
Status S=invokeCloudWatch()
ArrayListpreList= BootStap.getALLpeer()
ArrayListnewPeer=newArraylist()
For i=0 to peerList.size() do
If peerlist.get(i).fails() then
Peer peer=new Peer()
Peer.loadMYSQL.BackUpFromRDS(peerlist.get(i))
newPeer.add(peer)
Bootstrap.setBlackList(peerList.get(i))
Else peerList.get(i).overloaded() then
Peer peer=new peer()
Peer.upscale(peerList.get(i))
Peer.clone(peerList.get(i)).getDB(i)
BootStrap.setBlackList(peerList.get(i))
Newpeer.add(peer)
BootStrap.removeAllPeersInBlackList()
BootStrap.addAllNewPeer(newPeer)
BootStrap.broadcastnetworkStatus()
Sleep t seconds
    
```

**C.)Normal peer:**

Each normal peer has some processes like data loading and data indexing. In normal peer there are two data flows first is an offline data flow and an online data flow. Offline dataflow local businesses upload their data to MySQL database and information regarding this data is send to bootstrap peers that is the indexing information regarding data tables.



In online data flow the queries fired by various peers are executed against particular peer node which is having the required data with the help of bootstrap peer which stores the information about data from all peers.

**Query Processing:-**

As we know the data uploaded is present on number of peers so if a peer wants to access data from another peers, it fires the query which goes to bootstrap peer which have a

metadata of all peers and then it get fired on the peer sharing this data.

**D.)Adaptive Query Processing:**

Adaptive query processing(AQP) as a solution to the problems of query optimization and execution across relational, text, or XML data, regardless of whether the data is accessed locally, from the Web, or in a continuous stream. The system also adopts two additional optimizations to speed up the query processing. First, each normal peer caches sufficient table index, column index, and range index entries in memory to speed up the search for data owner peers, instead of traversing the BATON structure. Second, for equip join queries, the system employs bloom join algorithm to reduce the volume of data transmitted through the network. For small jobs, the P2P engine performs better than the Map Reduce engine, as it does not incur initialization cost and database join algorithms have been well optimized. However, for large scale data analytic jobs, the Map Reduce engine is more scalable, as it does not incur excursive data replications. Based on the above mentioned cost models, we propose our adaptive query processing approach. When a query is submitted, the query planner retrieves related histogram and index information from the bootstrap node, analyzes the query and constructs a processing graph for the query. Then the costs of both the P2P engine and Map Reduce engine are predicted based on the histograms and runtime parameters of the cost models. The query planner compares the costs between two methods and executes the one with lower cost.

**Adaptive Query Processing Algorithm**

```

Input: Query Q
Output: Query configuration on a specific query engine
TableSet S ← Table Parser (Q);
Cost Cmin ← MAX_VALUE;
QueryPlan Target ← null;
QueryPlanSet QS ← ∅;
foreach Table T ∈ S do
    //Generate Processing Graphs
    Rooted on T
    GraphSet GS = GraphGen (T);
    //Iterate through all Processing
    Graph rooted on T
    foreach Graph G ∈ GS do
        QueryPlan P1=P2P PalnGen (G);
        QueryPlan P2=MapresdPlanGen (G);
        QS=QS ∪ {P1};
        QS=QS ∪ {P2};
    foreach QueryPlan P ∈ QS do
        if CostEst (P) < Cmin then
            Cmin= CostEst (P);
            Target=p;
    return Target;
    
```

**Access Control:-**

In this technique we are giving an ability to each business to control their data sharing to specific number of companies.

And also adding security to this module by generating and sharing public and private keys.

#### IV. CONCLUSION AND FUTURE SCOPE

The In this project, It provides economical, flexible and scalable solutions for corporate network applications. From this presentation we understand how to perform low level query in large amount of data using Best Peer++ technology. Best peer ++ is a promising solution for efficient data sharing within co-operate network. It can efficiently handle typical workloads in co-operate network.

We could incorporate the rate of source peers updating dynamic data factors into the data updates dissemination management policies. The prediction of updates can bring some benefits with the limited resource usage. Second, in our current work, we implement the application layered on unstructured decentralized P2P systems. We could deploy the system on the structured P2P system, like Chord or CAN. These structured P2P systems have the routing ability, which makes data location more efficient. Thus, Peer Cast could combine the structure P2P techniques into our applications. Furthermore, we consider incorporate hybrid P2P systems, super-peer architectures, like KaZaA, which combines the advantages of centralized servers, and the autonomous peers. Last but still important point, in our current setting, Peer Cast framework is lack of the fairness and cooperation incentive mechanism. We suppose that peer users are all in a friendly-cooperative manner.

#### REFERENCE

- [1].I. Tatarinov, Z.G. Ives, J. Madhavan, A.Y. Halevy, D. Suci, N.N.Dalvi, X. Dong, Y. Kadiyska, G. Miklau, and P. Mork, "The PiazzaPeer Data Management Project," SIGMOD Record, vol. 32, no. 3, pp. 47-52, 2003.
- [2].W.S. Ng, B.C. Ooi, K.-L. Tan, and A. Zhou, "PeerDB: A P2P-Based System for Distributed Data Sharing," Proc. 19th Int'l Conf. Data Eng., pp. 633-644, 2003.
- [3].R. Huebsch, J.M. Hellerstein, N. Lanham, B.T. Loo, S. Shenker, and I. Stoica, "Querying the Internet with PIER," Proc. 29th Int'l Conf. Very Large Data Bases, pp.321-332, 2003.
- [4].Saepio Technologies Inc., "The Enterprise Marketing Management Strategy Guide," White Paper, 2010.
- [5] Thusoo, J. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, "HIVE: A Warehousing Solution over a Map-Reduce Framework," Proc. VLDB Endowment, vol. 2, no. 2, pp. 1626-1629, 2009.
- [6] H.T. Vo, C. Chen, and B.C. Ooi, "Towards Elastic Transactional Cloud Storage with Range Query Support," Proc. VLDB Endowment, vol. 3, no. 1, pp. 506-517, 2010.
- [7] S. Wu, S. Jiang, B.C. Ooi, and K.-L. Tan, "Distributed Online Aggregation," Proc. VLDB Endowment, vol. 2, no. 1, pp. 443-454, 2009.
- [8] S. Wu, J. Li, B.C. Ooi, and K.-L. Tan, "Just-in-Time Query Retrieval over Partially Indexed Data on Structured P2P Overlays," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '08), pp. 279-290, 2008.
- [9] S. Wu, Q.H. Vu, J. Li, and K.-L. Tan, "Adaptive Multi-Join Query Processing in PDBMS," Proc. IEEE Int'l Conf. Data Eng. (ICDE '09), pp. 1239-1242, 2009.
- [10] P. Rodriguez-Gianolli, M. Garzetti, L. Jiang, A. Kementsietsidis, I. Kiringa, M. Masud, R.J. Miller, and J. Mylopoulos, "Data Sharing in the Hyperion Peer Database System," Proc. Int'l Conf. Very Large Data Bases, pp. 1291-1294, 2005.
- [11] J. Dittrich, J. Quijano-Ruiz, A. Jindal, Y. Kargin, V. Setty, and JSchad, "Hadoop++: Making a Yellow Elephant Run Like a Cheetah (without it Even Noticing)," Proc. VLDB Endowment, vol. 3, no. 1/2, pp. 515-529, 2010.